Stream: Internet Engineering Task Force (IETF)
RFC: 8683
Category: Informational
Published: November 2019
ISSN: 2070-1721
Author: J. Palet Martinez
The IPv6 Company

RFC 8683
Additional Deployment Guidelines for NAT64/464XLAT in Operator and Enterprise Networks

Abstract

This document describes how Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers (NAT64) (including 464XLAT) can be deployed in an IPv6 network -- whether it's cellular ISP, broadband ISP, or enterprise -- and the possible optimizations. This document also discusses issues to be considered when having IPv6-only connectivity, such as: a) DNS64, b) applications or devices that use literal IPv4 addresses or non-IPv6-compliant APIs, and c) IPv4-only hosts or applications.

Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Not all documents approved by the IESG are candidates for any level of Internet Standard; see Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc8683.

Copyright Notice

Table of Contents

1. Introduction

Stateful NAT64 [RFC6146] describes a stateful IPv6-to-IPv4 translation mechanism that allows IPv6-only hosts to communicate with IPv4-only servers using unicast UDP, TCP, or ICMP by means of IPv4 public address sharing among multiple IPv6-only hosts. Unless otherwise stated, references to NAT64 (function) in this document should be interpreted as Stateful NAT64.

The translation of the packet headers is done using the IP/ICMP translation algorithm defined in [RFC7915]; algorithmically translating the IPv4 addresses to IPv6 addresses, and vice versa, is done following [RFC6052].

DNS64 [RFC6147] is in charge of the synthesis of AAAA records from the A records, so it only works for applications making use of DNS. It was designed to avoid changes in both the IPv6-only hosts and the IPv4-only server, so they can use a NAT64 function. As discussed in Section 5.5 of [RFC6147], a security-aware and validating host has to perform the DNS64 function locally.

However, the use of NAT64 and/or DNS64 presents three drawbacks:

1. Because DNS64 [RFC6147] modifies DNS answers, and DNSSEC is designed to detect such modifications, DNS64 [RFC6147] may potentially break DNSSEC, depending on a number of factors such as the location of the DNS64 function (at a DNS server or validator, at the end host, ...), how it has been configured, if the end hosts are validating, etc.
2. Because of the need to use DNS64 [RFC6147] or an alternative "host/application built-in" mechanism for address synthesis, there may be an issue for NAT64 [RFC6146] because it doesn't work when IPv4 literal addresses or non-IPv6-compliant APIs are being used.
3. NAT64 alone was not designed to provide a solution for IPv4-only hosts or applications that are located within a network and connected to a service provider IPv6-only access link, as it was designed for a very specific scenario (see Section 2.1 of [RFC6144]).

The drawbacks discussed above may come into play if part of an enterprise network is connected to other parts of the same network or to third-party networks by means of IPv6-only connectivity. This is just an example that may apply to many other similar cases. All of them are deployment specific.

Accordingly, the use of "operator", "operator network", "service provider", and similar terms in this document are interchangeable with equivalent cases of enterprise networks; other cases may be similar as well. This may be also the case for "managed end-user networks".

Note that if all the hosts in a network were performing address synthesis, as described in Section 7.2 of [RFC6147], some of the drawbacks may not apply. However, it is unrealistic to expect that in today's world, considering the high number of devices and applications that aren't yet IPv6 enabled. In this document, the case in which all hosts provide synthesis will be considered only for specific scenarios that can guarantee it.

An analysis of stateful IPv4/IPv6 mechanisms is provided in [RFC6889].

This document looks into different possible NAT64 [RFC6146] deployment scenarios, including IPv4-IPv6-IPv4 (464 for short) and similar ones that were not documented in [RFC6144], such as 464XLAT [RFC6877] in operator (broadband and cellular) and enterprise networks; it provides guidelines to avoid operational issues.

This document also explores the possible NAT64 deployment scenarios (split in "known to work" and "known to work under special conditions"), providing a quick and generic comparison table among them. Then, the document describes the issues that an operator needs to understand, which will allow the best approach/scenario to be defined for each specific network case. A summary provides some recommendations and decision points. A section with clarifications on

the usage of this document for enterprise networks is also provided. Finally, Appendix A provides an example of a broadband deployment using 464XLAT and hints for a customer-side translator (CLAT) implementation.

[RFC7269] already provides information about NAT64 deployment options and experiences. This document and [RFC7269] are complementary; they both look into different deployment considerations. Furthermore, this document considers the updated deployment experience and newer standards.

The target deployment scenarios in this document may also be covered by other IPv4-as-a-Service (IPv4aaS) transition mechanisms. Note that this is true only for broadband networks; in the case of cellular networks, the only supported solution is the use of NAT64/464XLAT. So, it is out of scope of this document to provide a comparison among the different IPv4aaS transition mechanisms, which are analyzed in [IPv6-TRANSITION].

Consequently, this document should not be used as a guide for an operator or enterprise to decide which IPv4aaS is the best one for its own network. Instead, it should be used as a tool for understanding all the implications, including relevant documents (or even specific parts of them) for the deployment of NAT64/464XLAT and for facilitating the decision process regarding specific deployment details.

2. Requirements Language

The key words "**MUST**", "**MUST NOT**", "**REQUIRED**", "**SHALL**", "**SHALL NOT**", "**SHOULD**", "**SHOULD NOT**", "**RECOMMENDED**", "**NOT RECOMMENDED**", "**MAY**", and "**OPTIONAL**" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. NAT64 Deployment Scenarios

DNS64 (see Section 7 of [RFC6147]) provides three deployment scenarios, depending on the location of the DNS64 function. However, since the publication of that document, other deployment scenarios and NAT64 use cases need to be considered in actual networks, despite the fact that some of them were specifically ruled out by the original NAT64/DNS64 work.

Consequently, the perspective in this document is to broaden those scenarios and include a few new ones. However, in order to reduce the number of possible cases, we work under the assumption that the service provider wants to make sure that all the customers have a service without failures. This means considering the following assumptions for the worst possible case:

1. There are hosts that will be validating DNSSEC.
2. IPv4 literal addresses and non-IPv6-compliant APIs are being used.
3. There are IPv4-only hosts or applications beyond the IPv6-only link (e.g., tethering in cellular networks).

This document uses a common set of possible "participant entities":

1. An IPv6-only access network (IPv6).
2. An IPv4-only remote network/server/service (IPv4).
3. A NAT64 function (NAT64) in the service provider.
4. A DNS64 function (DNS64) in the service provider.
5. An external service provider offering the NAT64 function and/or the DNS64 function (extNAT64/extDNS64).
6. A 464XLAT customer-side translator (CLAT).

Note that the nomenclature used in parentheses is the one that, for short, will be used in the figures. Note: for simplicity, the boxes in the figures don't mean they are actually a single device; they represent one or more functions as located in that part of the network (i.e., a single box with NAT64 and DNS64 functions can actually be several devices, not just one).

The possible scenarios are split in two general categories:

1. Known to work.
2. Known to work under special conditions.

3.1. Known to Work

The scenarios in this category are known to work, as there are well-known existing deployments from different operators using them. Each one may have different pros and cons, and in some cases, the trade-offs may be acceptable for some operators.

3.1.1. Service Provider NAT64 with DNS64

In this scenario (Figure 1), the service provider offers both the NAT64 and DNS64 functions.

This is the most common scenario as originally considered by the designers of NAT64 [RFC6146] and DNS64 [RFC6147]; however, it may also have the implications related to the DNSSEC.

This scenario may also fail to solve the issues of IPv4 literal addresses, non-IPv6-compliant APIs, or IPv4-only hosts or applications behind the IPv6-only access network.

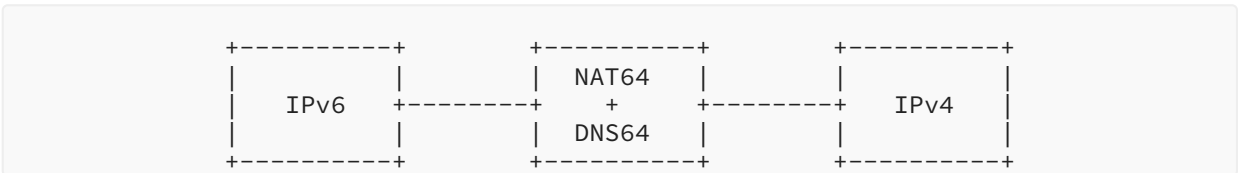

Figure 1: NAT64 with DNS64

A similar scenario (Figure 2) exists if the service provider offers only the DNS64 function; the NAT64 function is provided by an outsourcing agreement with an external provider. All the considerations in the previous paragraphs of this section are the same for this sub-case.

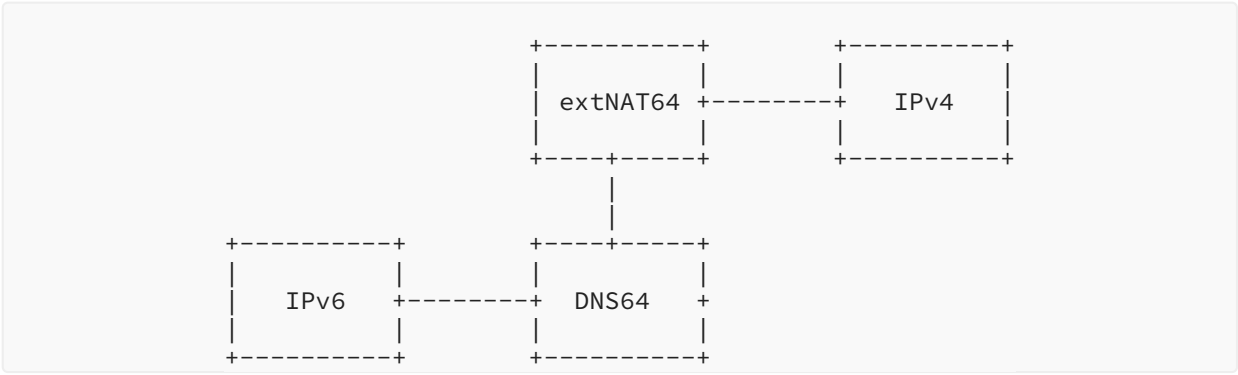

Figure 2: NAT64 in an External Service Provider

This is equivalent to the scenario (Figure 3) where the outsourcing agreement with the external provider is to provide both the NAT64 and DNS64 functions. Once more, all the considerations in the previous paragraphs of this section are the same for this sub-case.

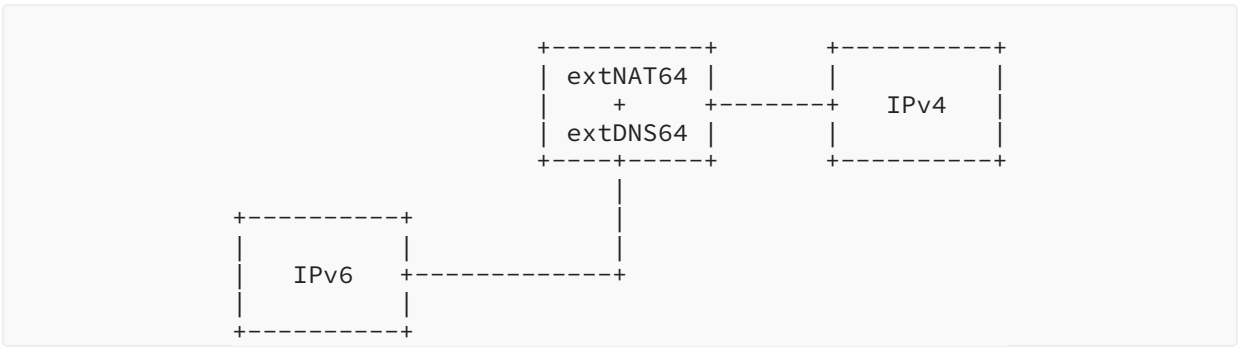

Figure 3: NAT64 and DNS64 in an External Provider

One additional equivalent scenario (Figure 4) exists if the service provider only offers the NAT64 function; the DNS64 function is from an external provider with or without a specific agreement among them. This is a common scenario today, as several "global" service providers provide free DNS/DNS64 services, and users often configure their DNS manually. This will only work if both the NAT64 and DNS64 functions are using the Well-Known Prefix (WKP) or the same Network-Specific Prefix (NSP). All the considerations in the previous paragraphs of this section are the same for this sub-case.

Of course, if the external DNS64 function is agreed with the service provider, then this case is similar to the ones already depicted in this scenario.

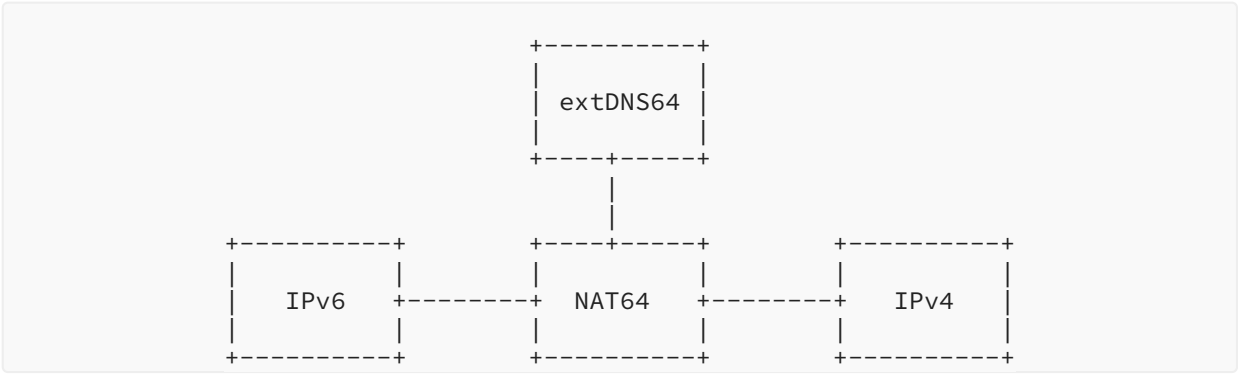

Figure 4: NAT64; DNS64 by an External Provider

3.1.2. Service Provider Offering 464XLAT Using DNS64

464XLAT [RFC6877] describes an architecture that provides IPv4 connectivity across a network, or part of it, when it is only natively transporting IPv6. The need to support the CLAT function in order to ensure the IPv4 service continuity in IPv6-only cellular deployments has been suggested in [RFC7849].

In order to do that, 464XLAT [RFC6877] relies on the combination of existing protocols:

1. The CLAT is a stateless IPv4-to-IPv6 translator (NAT46) [RFC7915] implemented in the end-user device or Customer Edge Router (CE), located at the "customer edge" of the network.
2. The provider-side translator (PLAT) is a stateful NAT64 [RFC6146], implemented typically in the operator network.
3. Optionally, DNS64 [RFC6147] may allow an optimization: a single translation at the NAT64, instead of two translations (NAT46+NAT64), when the application at the end-user device supports IPv6 DNS (uses AAAA Resource Records).

Note that even if the provider-side translator is referred to as PLAT in the 464XLAT terminology [RFC6877], for simplicity and uniformity across this document, it is always referred to as NAT64 (function).

In this scenario (Figure 5), the service provider deploys 464XLAT with a DNS64 function.

As a consequence, the DNSSEC issues remain, unless the host is doing the address synthesis.

464XLAT [RFC6877] is a very simple approach to cope with the major NAT64+DNS64 drawback: not working with applications or devices that use literal IPv4 addresses or non-IPv6-compliant APIs.

464XLAT [RFC6877] has been used mainly in IPv6-only cellular networks. By supporting a CLAT function, end-user device applications can access IPv4-only end networks / applications, despite the fact that those applications or devices use literal IPv4 addresses or non-IPv6-compliant APIs.

In addition, in the cellular network example above, if the User Equipment (UE) provides tethering, other devices behind it will be presented with a traditional Network Address Translation from IPv4 to IPv4 (NAT44), in addition to the native IPv6 support, so clearly it allows IPv4-only hosts behind the IPv6-only access network.

Furthermore, as discussed in [RFC6877], 464XLAT can be used in broadband IPv6 network architectures, by implementing the CLAT function at the CE.

The support of this scenario in a network offers two additional advantages:

- DNS load optimization: A CLAT should implement a DNS proxy (per [RFC5625]) so that only IPv6-native queries and AAAA records are sent to the DNS64 server. Otherwise, doubling the number of queries may impact the DNS infrastructure.
- Connection establishment delay optimization: If the UE/CE implementation is detecting the presence of a DNS64 function, it may issue only the AAAA query, instead of both the AAAA and A queries.

In order to understand all the communication possibilities, let's assume the following representation of two dual-stack (DS) peers:

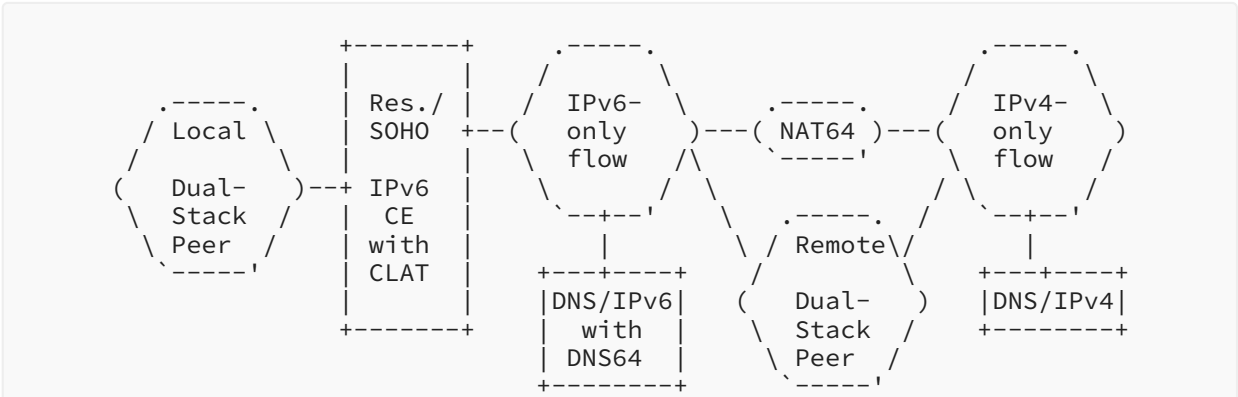

Figure A: Representation of 464XLAT among Two Peers with DNS64

In this case, the possible communication paths, among the IPv4/IPv6 stacks of both peers, are as follows:

1. Local-IPv6 to Remote-IPv6: Regular DNS and native IPv6 among peers.
2. Local-IPv6 to Remote-IPv4: DNS64 and NAT64 translation.
3. Local-IPv4 to Remote-IPv6: Not possible unless the CLAT implements Explicit Address Mappings (EAMs) as indicated by Section 4.9. In principle, it is not expected that services are deployed in the Internet when using IPv6 only, unless there is certainty that peers will also be IPv6 capable.
4. Local-IPv4 to Remote-IPv4: DNS64, CLAT, and NAT64 translations.
5. Local-IPv4 to Remote-dual-stack using EAM optimization: If the CLAT implements EAM as indicated by Section 4.9, instead of using the path d. above, NAT64 translation is avoided, and the flow will use IPv6 from the CLAT to the destination.

The rest of the figures in this section show different choices for placing the different elements.

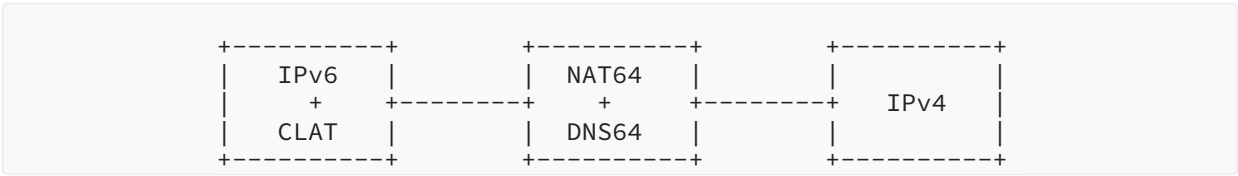

Figure 5: 464XLAT with DNS64

A similar scenario (Figure 6) exists if the service provider only offers the DNS64 function; the NAT64 function is provided by an outsourcing agreement with an external provider. All the considerations in the previous paragraphs of this section are the same for this sub-case.

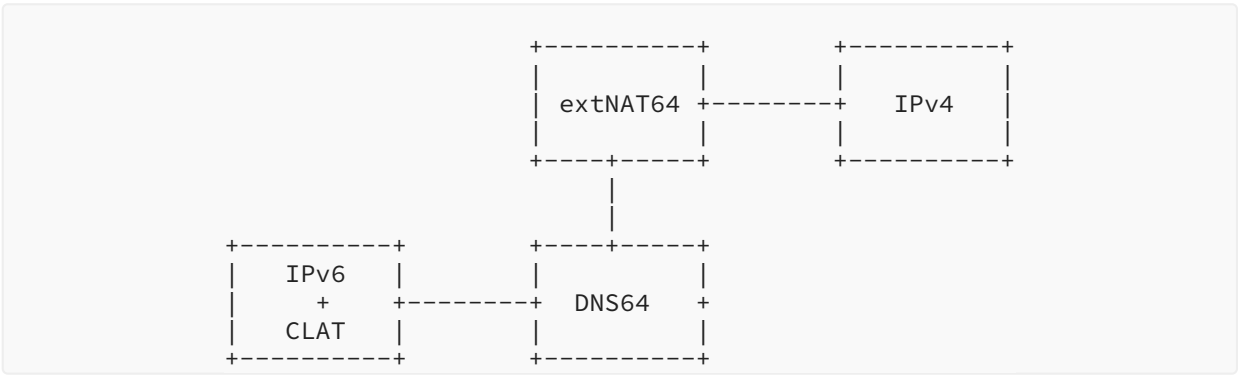

Figure 6: 464XLAT with DNS64; NAT64 in an External Provider

In addition, it is equivalent to the scenario (Figure 7) where the outsourcing agreement with the external provider is to provide both the NAT64 and DNS64 functions. Once more, all the considerations in the previous paragraphs of this section are the same for this sub-case.

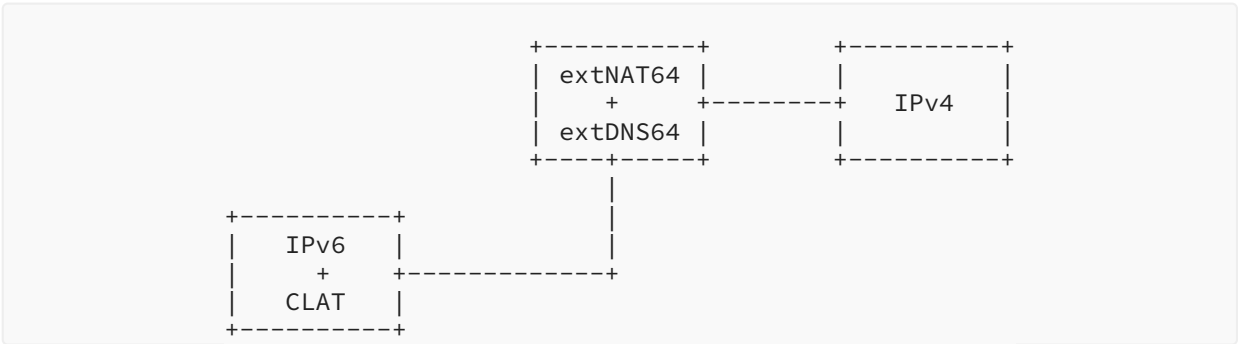

Figure 7: 464XLAT with DNS64; NAT64 and DNS64 in an External Provider

3.1.3. Service Provider Offering 464XLAT, without Using DNS64

The major advantage of this scenario (Figure 8), using 464XLAT without DNS64, is that the service provider ensures that DNSSEC is never broken, even if the user modifies the DNS configuration. Nevertheless, some CLAT implementations or applications may impose an extra delay, which is induced by the dual A/AAAA queries (and the wait for both responses), unless Happy Eyeballs v2 [RFC8305] is also present.

A possible variation of this scenario is when DNS64 is used only for the discovery of the NAT64 prefix. In the rest of the document, it is not considered a different scenario because once the prefix has been discovered, the DNS64 function is not used, so it behaves as if the DNS64 synthesis function is not present.

In this scenario, as in the previous one, there are no issues related to IPv4-only hosts (or IPv4-only applications) behind the IPv6-only access network, as neither are related to the usage of IPv4 literals or non-IPv6-compliant APIs.

The support of this scenario in a network offers one advantage:

- DNS load optimization: A CLAT should implement a DNS proxy (per [RFC5625]) so that only IPv6 native queries are sent to the DNS64 server. Otherwise, doubling the number of queries may impact the DNS infrastructure.

As indicated earlier, the connection establishment delay optimization is achieved only in the case of devices, Operating Systems, or applications that use Happy Eyeballs v2 [RFC8305], which is very common.

As in the previous case, let's assume the representation of two dual-stack peers:

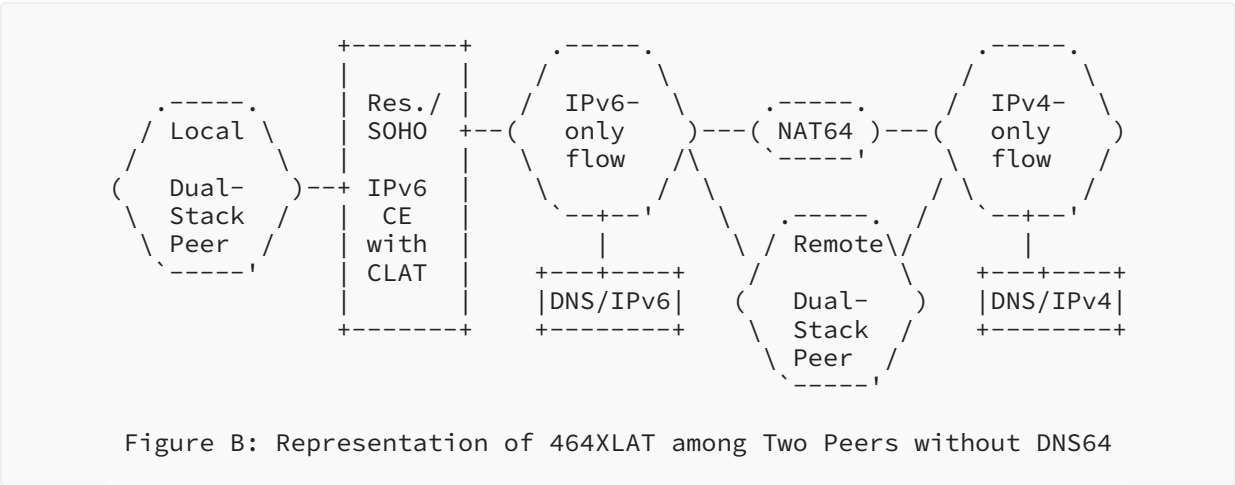

Figure B: Representation of 464XLAT among Two Peers without DNS64

In this case, the possible communication paths, among the IPv4/IPv6 stacks of both peers, are as follows:

1. Local-IPv6 to Remote-IPv6: Regular DNS and native IPv6 among peers.
2. Local-IPv6 to Remote-IPv4: Regular DNS, CLAT, and NAT64 translations.

3. Local-IPv4 to Remote-IPv6: Not possible unless the CLAT implements EAM as indicated by Section 4.9. In principle, it is not expected that services are deployed in the Internet using IPv6 only, unless there is certainty that peers will also be IPv6-capable.
4. Local-IPv4 to Remote-IPv4: Regular DNS, CLAT, and NAT64 translations.
5. Local-IPv4 to Remote-dual-stack using EAM optimization: If the CLAT implements EAM as indicated by Section 4.9, instead of using the path d. above, NAT64 translation is avoided, and the flow will use IPv6 from the CLAT to the destination.

Notice that this scenario works while the local hosts/applications are dual stack (which is the current situation) because the connectivity from a local IPv6 to a remote IPv4 is not possible without a AAAA synthesis. This aspect is important only when there are IPv6-only hosts in the LANs behind the CLAT and they need to communicate with remote IPv4-only hosts. However, it is not a sensible approach from an Operating System or application vendor perspective to provide IPv6-only support unless, similar to case c above, there is certainty of peers supporting IPv6 as well. An approach to a solution for this is also presented in [OPT-464XLAT].

The following figures show different choices for placing the different elements.

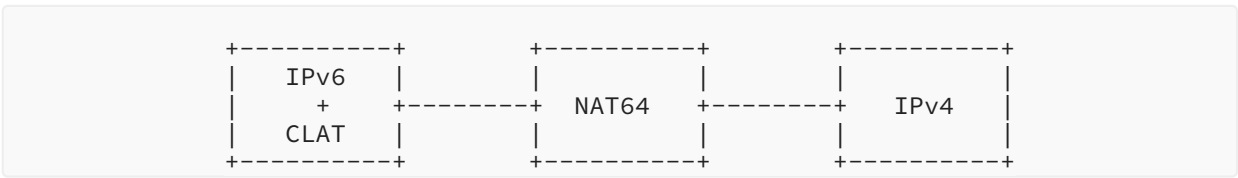

Figure 8: 464XLAT without DNS64

This is equivalent to the scenario (Figure 9) where there is an outsourcing agreement with an external provider for the NAT64 function. All the considerations in the previous paragraphs of this section are the same for this sub-case.

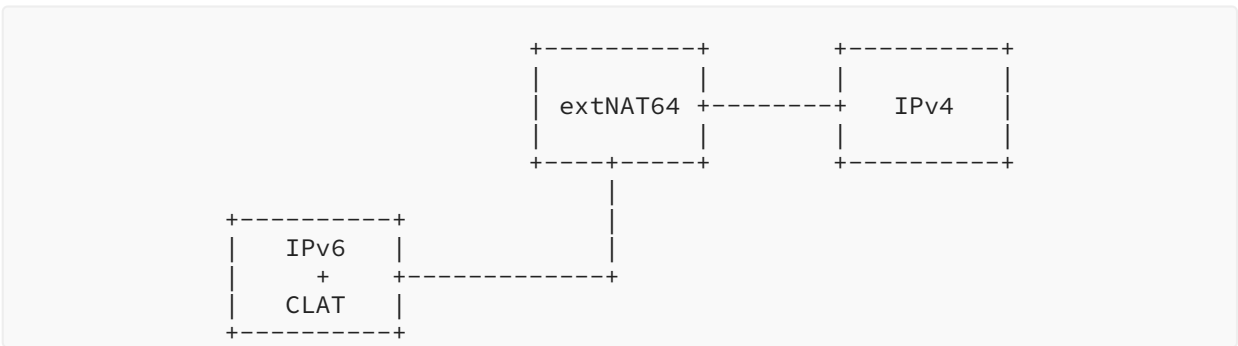

Figure 9: 464XLAT without DNS64; NAT64 in an External Provider

3.2. Known to Work under Special Conditions

The scenarios in this category are known not to work unless significant effort is devoted to solving the issues or they are intended to solve problems across "closed" networks instead of as a general Internet access usage. Even though some of the different pros, cons, and trade-offs may be acceptable, operators have implementation difficulties, as their expectations of NAT64/DNS64 are beyond the original intent.

3.2.1. Service Provider NAT64 without DNS64

In this scenario (Figure 10), the service provider offers a NAT64 function; however, there is no DNS64 function support at all.

As a consequence, an IPv6 host in the IPv6-only access network will not be able to detect the presence of DNS64 by means of [RFC7050] or learn the IPv6 prefix to be used for the NAT64 function.

This can be sorted out as indicated in Section 4.1.1.

Regardless, because of the lack of the DNS64 function, the IPv6 host will not be able to obtain AAAA synthesized records, so the NAT64 function becomes useless.

An exception to this "useless" scenario is to manually configure mappings between the A records of each of the IPv4-only remote hosts and the corresponding AAAA records with the WKP or NSP used by the service-provider NAT64 function, as if they were synthesized by a DNS64 function.

This mapping could be done by several means, typically at the authoritative DNS server or at the service-provider resolvers by means of DNS Response Policy Zones (RPZs) [DNS-RPZ] or equivalent functionality. DNS RPZ may have implications in DNSSEC if the zone is signed. Also, if the service provider is using an NSP, having the mapping at the authoritative server may create troubles for other parties trying to use a different NSP or WKP, unless multiple DNS "views" (split-DNS) are also being used at the authoritative servers.

Generally, the mappings alternative will only make sense if a few sets of IPv4-only remote hosts need to be accessed by a single network (or a small number of them), which supports IPv6 only in the access. This will require some kind of mutual agreement for using this procedure; this should not be a problem because it won't interfere with Internet use (which is a "closed service").

In any case, this scenario doesn't solve the issue of IPv4 literal addresses, non-IPv6-compliant APIs, or IPv4-only hosts within that IPv6-only access network.

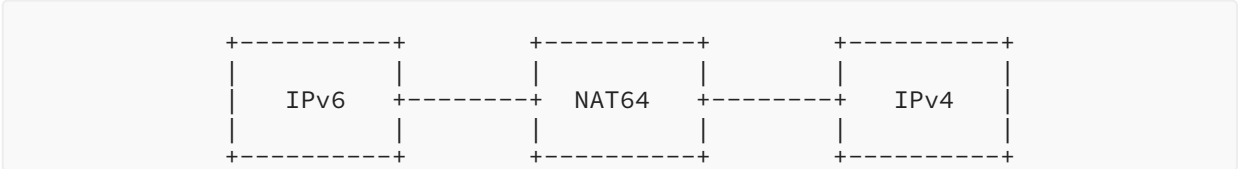

Figure 10: NAT64 without DNS64

3.2.2. Service-Provider NAT64; DNS64 in IPv6 Hosts

In this scenario (Figure 11), the service provider offers the NAT64 function but not the DNS64 function. However, the IPv6 hosts have a built-in DNS64 function.

This may become common if the DNS64 function is implemented in all the IPv6 hosts/stacks. This is not common at the time of writing but may become more common in the near future. This way, the DNSSEC validation is performed on the A record, and then the host can use the DNS64 function in order to use the NAT64 function without any DNSSEC issues.

This scenario fails to solve the issue of IPv4 literal addresses or non-IPv6-compliant APIs, unless the IPv6 hosts also support Happy Eyeballs v2 (Section 7.1 of [RFC8305]).

Moreover, this scenario also fails to solve the problem of IPv4-only hosts or applications behind the IPv6-only access network.

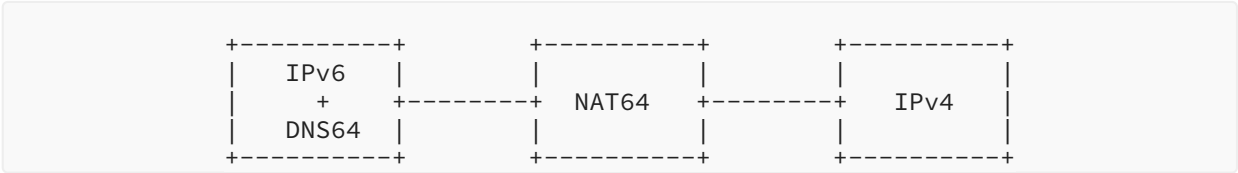

Figure 11: NAT64; DNS64 in IPv6 Hosts

3.2.3. Service-Provider NAT64; DNS64 in the IPv4-Only Remote Network

In this scenario (Figure 12), the service provider offers the NAT64 function only. The IPv4-only remote network offers the DNS64 function.

This is not common, and it doesn't make sense that a remote network, not deploying IPv6, is providing a DNS64 function. Like the scenario depicted in Section 3.2.1, it will only work if both sides are using the WKP or the same NSP, so the same considerations apply. It can also be tuned to behave as in Section 3.1.1.

This scenario fails to solve the issue of IPv4 literal addresses or non-IPv6-compliant APIs.

Moreover, this scenario also fails to solve the problem of IPv4-only hosts or applications behind the IPv6-only access network.

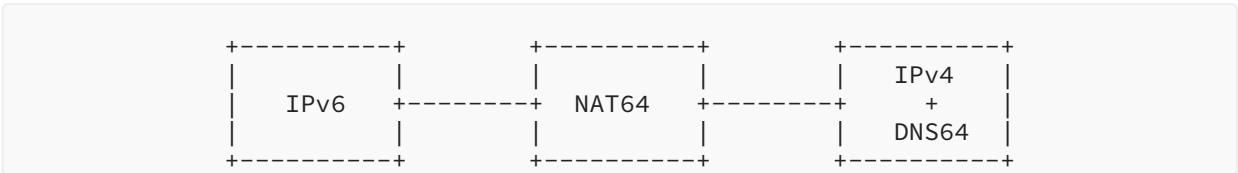

Figure 12: NAT64; DNS64 in IPv4-Only Hosts

3.3. Comparing the Scenarios

This section compares the different scenarios, including possible variations (each one represented in the previous sections by a different figure), while considering the following criteria:

1. DNSSEC: Are there hosts validating DNSSEC?
2. Literal/APIs: Are there applications using IPv4 literals or non-IPv6-compliant APIs?
3. IPv4 only: Are there hosts or applications using IPv4 only?
4. Foreign DNS: Does the scenario survive if the user, Operating System, applications, or devices change the DNS?
5. DNS load opt. (DNS load optimization): Are there extra queries that may impact the DNS infrastructure?
6. Connect. opt. (connection establishment delay optimization): Is the UE/CE only issuing the AAAA query or also the A query and waiting for both responses?

In the table below, the columns represent each of the scenarios from the previous sections by the figure number. The possible values are as follows:

"-" means the scenario is "bad" for that criterion.

"+" means the scenario is "good" for that criterion.

"*" means the scenario is "bad" for that criterion; however, it is typically resolved with the support of Happy Eyeballs v2 [RFC8305].

In some cases, "countermeasures", alternative or special configurations, may be available for the criterion designated as "bad". So, this comparison is considering a generic case as a quick comparison guide. In some cases, a "bad" criterion is not necessarily a negative aspect; it all depends on the specific needs/characteristics of the network where the deployment will take place. For instance, in a network that only has IPv6-only hosts and apps using DNS and IPv6-compliant APIs, there is no impact using only NAT64 and DNS64, but if the hosts validate DNSSEC, that criterion is still relevant.

Item / Figure	1	2	3	4	5	6	7	8	9	10	11	12
DNSSEC	-	-	-	-	-	-	-	+	+	+	+	+
Literal/APIs	-	-	-	-	+	+	+	+	+	-	-	-
IPv4-only	-	-	-	-	+	+	+	+	+	-	-	-
Foreign DNS	-	-	-	-	+	+	+	+	+	-	+	-
DNS load opt.	+	+	+	+	+	+	+	+	+	+	+	+
Connect. opt.	+	+	+	+	+	+	+	*	*	+	+	+

Table 1: Scenario Comparison

As a general conclusion, we should note if the network must support applications using any of the following:

- IPv4 literals
- non-IPv6-compliant APIs
- IPv4-only hosts or applications

Then, only the scenarios with 464XLAT, a CLAT function, or equivalent built-in local address synthesis features will provide a valid solution. Furthermore, those scenarios will also keep working if the DNS configuration is modified. Clearly, depending on if DNS64 is used or not, DNSSEC may be broken for those hosts doing DNSSEC validation.

All the scenarios are good in terms of DNS load optimization, and in the case of 464XLAT, it may provide an extra degree of optimization. Finally, all of the scenarios are also good in terms of connection establishment delay optimization. However, in the case of 464XLAT without DNS64, the usage of Happy Eyeballs v2 is required. This is not an issue as it is commonly available in actual Operating Systems.

4. Issues to be Considered

This section reviews the different issues that an operator needs to consider for a NAT64/464XLAT deployment, as they may develop specific decision points about how to approach that deployment.

4.1. DNSSEC Considerations and Possible Approaches

As indicated in the security considerations for DNS64 (see Section 8 of [RFC6147]) because DNS64 modifies DNS answers and DNSSEC is designed to detect such modifications, DNS64 may break DNSSEC.

When a device connected to an IPv6-only access network queries for a domain name in a signed zone, by means of a recursive name server that supports DNS64, the result may be a synthesized AAAA record. In that case, if the recursive name server is configured to perform DNSSEC validation and has a valid chain of trust to the zone in question, it will cryptographically validate the negative response from the authoritative name server. This is the expected DNS64 behavior: the recursive name server actually "lies" to the client device. However, in most of the cases, the client will not notice it, because generally, they don't perform validation themselves; instead, they rely on the recursive name servers.

In fact, a validating DNS64 resolver increases the confidence on the synthetic AAAA, as it has validated that a non-synthetic AAAA doesn't exist. However, if the client device is oblivious to NAT64 (the most common case) and performs DNSSEC validation on the AAAA record, it will fail as it is a synthesized record.

The best possible scenario from a DNSSEC point of view is when the client requests that the DNS64 server perform the DNSSEC validation (by setting the DNSSEC OK (DO) bit to 1 and the CD bit to 0). In this case, the DNS64 server validates the data; thus, tampering may only happen inside the DNS64 server (which is considered as a trusted part, thus, its likelihood is low) or between the DNS64 server and the client. All other parts of the system (including transmission and caching) are protected by DNSSEC [Threat-DNS64].

Similarly, if the client querying the recursive name server is another name server configured to use it as a forwarder, and it is performing DNSSEC validation, it will also fail on any synthesized AAAA record.

All those considerations are extensively covered in Sections 3, 5.5, and 6.2 of [RFC6147].

DNSSEC issues could be avoided if all the signed zones provide IPv6 connectivity together with the corresponding AAAA records. However, this is out of the control of the operator needing to deploy a NAT64 function. This has been proposed already in [DNS-DNSSEC].

An alternative solution, which was considered while developing [RFC6147], is that the validators will be DNS64 aware. Then, they can perform the necessary discovery and do their own synthesis. Since that was standardized sufficiently early in the validator deployment curve, the expectation was that it would be okay to break certain DNSSEC assumptions for networks that were stuck and really needing NAT64/DNS64.

As already indicated, the scenarios in the previous section are simplified to look at the worst possible case and for the most perfect approach. A DNSSEC breach will not happen if the end host is not doing validation.

The figures in previous studies indicate that DNSSEC broken by using DNS64 makes up about 1.7% [About-DNS64] of the cases. However, we can't negate that this may increase as DNSSEC deployment grows. Consequently, a decision point for the operator must depend on the following question: Do I really care about that percentage of cases and the impact on my help desk, or can I provide alternative solutions for them? Some possible solutions may be exist, as depicted in the next sections.

4.1.1. Not Using DNS64

One solution is to avoid using DNS64, but as already indicated, this is not possible in all the scenarios.

The use of DNS64 is a key component for some networks, in order to comply with traffic performance metrics, monitored by some governmental bodies and other institutions [FCC] [ARCEP].

One drawback of not having a DNS64 on the network side is that it's not possible to heuristically discover NAT64 [RFC7050]. Consequently, an IPv6 host behind the IPv6-only access network will not be able to detect the presence of the NAT64 function, nor learn the IPv6 prefix to be used for it, unless it is configured by alternative means.

The discovery of the IPv6 prefix could be solved, as described in [RFC7050], by means of adding the relevant AAAA records to the ipv4only.arpa. zone of the service-provider recursive servers, i.e., if using the WKP (64:ff9b::/96):

```
ipv4only.arpa.  SOA     . . 0 0 0 0 0
ipv4only.arpa.  NS      .
ipv4only.arpa.  AAAA    64:ff9b::192.0.0.170
ipv4only.arpa.  AAAA    64:ff9b::192.0.0.171
ipv4only.arpa.  A       192.0.0.170
ipv4only.arpa.  A       192.0.0.171
```

An alternative option is the use of DNS RPZ [DNS-RPZ] or equivalent functionalities. Note that this may impact DNSSEC if the zone is signed.

Another alternative, only valid in environments with support from the Port Control Protocol (PCP) (for both the hosts or CEs and for the service-provider network), is to follow "Discovering NAT64 IPv6 Prefixes Using the Port Control Protocol (PCP)" [RFC7225].

Other alternatives may be available in the future. All them are extensively discussed in [RFC7051]; however, due to the deployment evolution, many considerations from that document have changed. New options are being documented, such as using Router Advertising [PREF64] or DHCPv6 options [DHCPv6-OPTIONS].

Simultaneous support of several of the possible approaches is convenient and will ensure that clients with different ways to configure the NAT64 prefix successfully obtain it. This is also convenient even if DNS64 is being used.

Also of special relevance to this section is [IPV4ONLY-ARPA].

4.1.2. DNSSEC Validator Aware of DNS64

In general, by default, DNS servers with DNS64 function will not synthesize AAAA responses if the DO flag was set in the query.

In this case, since only an A record is available, if a CLAT function is present, the CLAT will, as in the case of literal IPv4 addresses, keep that traffic flow end to end as IPv4 so DNSSEC is not broken.

However, this will not work if a CLAT function is not present because the hosts will not be able to use IPv4 (which is the case for all the scenarios without 464XLAT).

4.1.3. Stub Validator

If the DO flag is set and the client device performs DNSSEC validation, and the Checking Disabled (CD) flag is set for a query, the DNS64 recursive server will not synthesize AAAA responses. In this case, the client could perform the DNSSEC validation with the A record and then synthesize the AAAA responses [RFC6052]. For that to be possible, the client must have learned the NAT64 prefix beforehand using any of the available methods (see [RFC7050], [RFC7225], [PREF64], and [DHCPv6-OPTIONS]). This allows the client device to avoid using the DNS64 function and still use NAT64 even with DNSSEC.

If the end host is IPv4 only, this will not work if a CLAT function is not present (which is the case for all scenarios without 464XLAT).

Instead of a CLAT, some devices or Operating Systems may implement an equivalent function by using Bump-in-the-Host [RFC6535] as part of Happy Eyeballs v2 (see Section 7.1 of [RFC8305]). In this case, the considerations in the above paragraphs are also applicable.

4.1.4. CLAT with DNS Proxy and Validator

If a CE includes CLAT support and also a DNS proxy, as indicated in Section 6.4 of [RFC6877], the CE could behave as a stub validator on behalf of the client devices. Then, following the same approach described in Section 4.1.3, the DNS proxy will actually "lie" to the client devices, which, in most cases, will not be noticed unless they perform validation by themselves. Again, this allows the client devices to avoid the use of the DNS64 function but to still use NAT64 with DNSSEC.

Once more, this will not work without a CLAT function (which is the case for all scenarios without 464XLAT).

4.1.5. ACL of Clients

In cases of dual-stack clients, AAAA queries typically take preference over A queries. If DNS64 is enabled for those clients, it will never get A records, even for IPv4-only servers.

As a consequence, in cases where there are IPv4-only servers, and those are located in the path before the NAT64 function, the clients will not be able to reach them. If DNSSEC is being used for all those flows, specific addresses or prefixes can be left out of the DNS64 synthesis by means of Access Control Lists (ACLs).

Once more, this will not work without a CLAT function (which is the case for all scenarios without 464XLAT).

4.1.6. Mapping Out IPv4 Addresses

If there are well-known specific IPv4 addresses or prefixes using DNSSEC, they can be mapped out of the DNS64 synthesis.

Even if this is not related to DNSSEC, this "mapping-out" feature is quite commonly used to ensure that addresses [RFC1918] (for example, used by LAN servers) are not synthesized to AAAA.

Once more, this will not work without a CLAT function (which is the case for all scenarios without 464XLAT).

4.2. DNS64 and Reverse Mapping

When a client device using DNS64 tries to reverse-map a synthesized IPv6 address, the name server responds with a CNAME record that points the domain name used to reverse-map the synthesized IPv6 address (the one under ip6.arpa) to the domain name corresponding to the embedded IPv4 address (under in-addr.arpa).

This is the expected behavior, so no issues need to be considered regarding DNS reverse mapping.

4.3. Using 464XLAT with/without DNS64

In case the client device is IPv6 only (either because the stack or application is IPv6 only or because it is connected via an IPv6-only LAN) and the remote server is IPv4 only (either because the stack is IPv4 only or because it is connected via an IPv4-only LAN), only NAT64 combined with DNS64 will be able to provide access between both. Because DNS64 is then required, DNSSEC validation will only be possible if the recursive name server is validating the negative response from the authoritative name server, and the client is not performing validation.

Note that at this stage of the transition, it is not expected that applications, devices, or Operating Systems are IPv6 only. It will not be a sensible decision for a developer to work on that direction, unless it is clear that the deployment scenario fully supports it.

On the other hand, an end user or enterprise network may decide to run IPv6 only in the LANs. In case there is any chance for applications to be IPv6 only, the Operating System may be responsible for either doing a local address synthesis or setting up some kind of on-demand VPN (IPv4-in-IPv6), which needs to be supported by that network. This may become very common in enterprise networks, where "Unique IPv6 Prefix per Host" [RFC8273] is supported.

However, when the client device is dual stack and/or connected in a dual-stack LAN by means of a CLAT function (or has a built-in CLAT function), DNS64 is an option.

1. With DNS64: If DNS64 is used, most of the IPv4 traffic (except if using literal IPv4 addresses or non-IPv6-compliant APIs) will not use the CLAT and will instead use the IPv6 path, so only one translation will be done at the NAT64. This may break DNSSEC, unless measures as described in the previous sections are taken.

2. Without DNS64: If DNS64 is not used, all the IPv4 traffic will make use of the CLAT, so two translations are required (NAT46 at the CLAT and NAT64 at the PLAT), which adds some overhead in terms of the extra NAT46 translation. However, this avoids the AAAA synthesis and consequently will never break DNSSEC.

Note that the extra translation, when DNS64 is not used, takes place at the CLAT, which means no extra overhead for the operator. However, it adds potential extra delays to establish the connections and has no perceptible impact for a CE in a broadband network, but it may have some impact on a battery-powered device. The cost for a battery-powered device is possibly comparable to the cost when the device is doing a local address synthesis (see Section 7.1 of [RFC8305]).

4.4. Foreign DNS

Clients, devices, or applications in a service-provider network may use DNS servers from other networks. This may be the case if individual applications use their own DNS server, the Operating System itself or even the CE, or combinations of the above.

Those "foreign" DNS servers may not support DNS64; as a consequence, those scenarios that require a DNS64 may not work. However, if a CLAT function is available, the considerations in Section 4.3 will apply.

If the foreign DNS supports the DNS64 function, incorrect configuration parameters may be provided that, for example, cause WKP or NSP to become unmatched or result in a case such as the one described in Section 3.2.3.

Having a CLAT function, even if using foreign DNS without a DNS64 function, ensures that everything will work, so the CLAT must be considered to be an advantage despite user configuration errors. As a result, all the traffic will use a double translation (NAT46 at the CLAT and NAT64 at the operator network), unless there is support for EAM (Section 4.9).

An exception is the case where there is a CLAT function at the CE that is not able to obtain the correct configuration parameters (again, causing WKP or NSP to become unmatched).

However, it needs to be emphasized that if there is no CLAT function (which is the case for all scenarios without 464XLAT), an external DNS without DNS64 support will disallow any access to IPv4-only destination networks and will not guarantee the correct DNSSEC validation, so it will behave as in Section 3.2.1.

In summary, the consequences of using foreign DNS depends on each specific case. However, in general, if a CLAT function is present, most of the time there will not be any issues. In the other cases, the access to IPv6-enabled services is still guaranteed for IPv6-enabled hosts, but it is not guaranteed for IPv4-only hosts nor is the access to IPv4-only services for any hosts in the network.

The causes of "foreign DNS" could be classified in three main categories, as depicted in the following subsections.

4.4.1. Manual Configuration of DNS

It is becoming increasingly common that end users, or even devices or applications, configure alternative DNS in their Operating Systems and sometimes in CEs.

4.4.2. DNS Privacy/Encryption Mechanisms

Clients or applications may use mechanisms for DNS privacy/encryption, such as DNS over TLS (DoT) [RFC7858], DNS over DTLS [RFC8094], DNS queries over HTTPS (DoH) [RFC8484], or DNS over QUIC (DoQ) [QUIC-CONNECTIONS].

Currently, those DNS privacy/encryption options are typically provided by the applications, not the Operating System vendors. At the time this document was written, the DoT and DoH standards have declared DNS64 (and consequently NAT64) out of their scope, so an application using them may break NAT64, unless a correctly configured CLAT function is used.

4.4.3. Split DNS and VPNs

When networks or hosts use "split-DNS" (also called Split Horizon, DNS views, or private DNS), the successful use of DNS64 is not guaranteed. This case is analyzed in Section 4 of [RFC6950].

A similar situation may happen with VPNs that force all the DNS queries through the VPN and ignore the operator DNS64 function.

4.5. Well-Known Prefix (WKP) vs. Network-Specific Prefix (NSP)

Section 3 of "IPv6 Addressing of IPv4/IPv6 Translator" [RFC6052] discusses some considerations that are useful to an operator when deciding if a WKP or an NSP should be used.

Considering that discussion and other issues, we can summarize the possible decision points to as follows:

1. The WKP **MUST NOT** be used to represent non-global IPv4 addresses. If this is required because the network to be translated uses non-global addresses, then an NSP is required.
2. The WKP **MAY** appear in interdomain routing tables, if the operator provides a NAT64 function to peers. However, in this case, special considerations related to BGP filtering are required, and IPv4-embedded IPv6 prefixes longer than the WKP **MUST NOT** be advertised (or accepted) in BGP. An NSP may be a more appropriate option in those cases.
3. If several NAT64s use the same prefix, packets from the same flow may be routed to a different NAT64 in case of routing changes. This can be avoided by either using different prefixes for each NAT64 function or ensuring that all the NAT64s coordinate their state. Using an NSP could simplify that.
4. If DNS64 is required and users, devices, Operating Systems, or applications may change their DNS configuration and deliberately choose an alternative DNS64 function, the alternative DNS64 will most likely use the WKP by default. In that case, if an NSP is used by the NAT64 function, clients will not be able to use the operator NAT64 function, which will break connectivity to IPv4-only destinations.

4.6. IPv4 Literals and Non-IPv6-Compliant APIs

A host or application using literal IPv4 addresses or older APIs, which aren't IPv6 compliant, behind a network with IPv6-only access will not work unless any of the following alternatives are provided:

- CLAT (or an equivalent function).
- Happy Eyeballs v2 (Section 7.1 of [RFC8305]).
- Bump-in-the-Host [RFC6535] with a DNS64 function.

Those alternatives will solve the problem for an end host. However, if the end host is providing "tethering" or an equivalent service to other hosts, that needs to be considered as well. In other words, in a cellular network, these alternatives resolve the issue for the UE itself, but this may not be the case for hosts connected via the tethering.

Otherwise, the support of 464XLAT is the only valid and complete approach to resolve this issue.

4.7. IPv4-Only Hosts or Applications

IPv4-only hosts or an application behind a network with IPv6-only access will not work unless a CLAT function is present.

464XLAT is the only valid approach to resolve this issue.

4.8. CLAT Translation Considerations

As described in "IPv6 Prefix Handling" (see Section 6.3 of [RFC6877]), if the CLAT function can be configured with a dedicated /64 prefix for the NAT46 translation, then it will be possible to do a more efficient stateless translation.

Otherwise, if this dedicated prefix is not available, the CLAT function will need to do a stateful translation, for example, perform stateful NAT44 for all the IPv4 LAN packets so they appear as coming from a single IPv4 address; in turn, the CLAT function will perform a stateless translation to a single IPv6 address.

A possible setup, in order to maximize the CLAT performance, is to configure the dedicated translation prefix. This can be easily achieved automatically, if the broadband CE or end-user device is able to obtain a shorter prefix by means of DHCPv6-PD [RFC8415] or other alternatives. The CE can then use a specific /64 for the translation. This is also possible when broadband is provided by a cellular access.

The above recommendation is often not possible for cellular networks, when connecting smartphones (as UEs): generally they don't use DHCPv6-PD [RFC8415]. Instead, a single /64 is provided for each Packet Data Protocol (PDP) context, and prefix sharing [RFC6877] is used. In this case, the UEs typically have a build-in CLAT function that is performing a stateful NAT44 translation before the stateless NAT46.

4.9. EAM Considerations

"Explicit Address Mappings for Stateless IP/ICMP Translation" [RFC7757] provides a way to configure explicit mappings between IPv4 and IPv6 prefixes of any length. When this is used, for example, in a CLAT function, it may provide a simple mechanism in order to avoid traffic flows between IPv4-only nodes or applications and dual-stack destinations to be translated twice (NAT46 and NAT64), by creating mapping entries with the Global Unicast Address (GUA) of the IPv6-reachable destination. This optimization of NAT64 usage is very useful in many scenarios, including Content Delivery Networks (CDNs) and caches, as described in [OPT-464XLAT].

In addition, it may also provide a way for IPv4-only nodes or applications to communicate with IPv6-only destinations.

4.10. Incoming Connections

The use of NAT64, in principle, disallows IPv4 incoming connections, which may still be needed for IPv4-only peer-to-peer applications. However, there are several alternatives that resolve this issue:

1. Session Traversal Utilities for NAT (STUN) [RFC5389], Traversal Using Relays around NAT (TURN) [RFC5766], and Interactive Connectivity Establishment (ICE) [RFC8445] are commonly used by peer-to-peer applications in order to allow incoming connections with IPv4 NAT. In the case of NAT64, they work as well.
2. The Port Control Protocol (PCP) [RFC6887] allows a host to control how incoming IPv4 and IPv6 packets are translated and forwarded. A NAT64 may implement PCP to allow this service.
3. EAM [RFC7757] may also be used in order to configure explicit mappings for customers that require them. This is used, for example, by Stateless IP/ICMP Translation for IPv6 Data Center Environments (SIIT-DC) [RFC7755] and SIIT-DC Dual Translation Mode (SIIT-DC-DTM) [RFC7756].

5. Summary of Deployment Recommendations for NAT64/464XLAT

It has been demonstrated that NAT64/464XLAT is a valid choice in several scenarios (IPv6-IPv4 and IPv4-IPv6-IPv4), being the predominant mechanism in the majority of the cellular networks, which account for hundreds of millions of users [ISOC]. NAT64/464XLAT offer different choices of deployment, depending on each network case, needs, and requirements. Despite that, this document is not an explicit recommendation for using this choice versus other IPv4aaS transition mechanisms. Instead, this document is a guide that facilitates evaluating a possible implementation of NAT64/464XLAT and key decision points about specific design considerations for its deployment.

Depending on the specific requirements of each deployment case, DNS64 may be a required function, while in other cases, the adverse effects may be counterproductive. Similarly, in some cases, a NAT64 function, together with a DNS64 function, may be a valid solution when there is a

certainty that IPv4-only hosts or applications do not need to be supported (see Sections 4.6 and 4.7). However, in other cases (i.e., IPv4-only devices or applications that need to be supported), the limitations of NAT64/DNS64 may indicate that the operator needs to look into 464XLAT as a more complete solution.

For broadband-managed networks (where the CE is provided or suggested/supported by the operator), in order to fully support the actual user's needs (i.e., IPv4-only devices and applications and the usage of IPv4 literals and non-IPv6-compliant APIs), the 464XLAT scenario should be considered. In that case, it must support a CLAT function.

If the operator provides DNS services, they may support a DNS64 function to avoid, as much as possible, breaking DNSSEC. This will also increase performance, by reducing the double translation for all the IPv4 traffic. In this case, if the DNS service is offering DNSSEC validation, then it must be in such a way that it is aware of the DNS64. This is considered the simpler and safer approach, and it may be combined with other recommendations described in this document:

- DNS infrastructure **MUST** be aware of DNS64 (Section 4.1.2).
- Devices running CLAT **SHOULD** follow the indications in "Stub Validator" (see Section 4.1.3). However, this may be out of the control of the operator.
- CEs **SHOULD** include a DNS proxy and validator (Section 4.1.4).
- "ACL of Clients" (see Section 4.1.5) and "Mapping Out IPv4 Addresses" (see Section 4.1.6) **MAY** be considered by operators, depending on their own infrastructure.

This "increased performance" approach has the disadvantage of potentially breaking DNSSEC for a small percentage of validating end hosts versus the small impact of a double translation taking place in the CE. If CE performance is not an issue, which is the most frequent case, then a much safer approach is to not use DNS64 at all, and consequently, ensure that all the IPv4 traffic is translated at the CLAT (Section 4.3).

If DNS64 is not used, at least one of the alternatives described in Section 4.1.1 must be followed in order to learn the NAT64 prefix.

The operator needs to consider that if the DNS configuration is modified (see Sections 4.4, 4.4.2, and 4.4.3), which most likely cannot be avoided, a foreign non-DNS64 could be used instead of configuring a DNS64. In a scenario with only a NAT64 function, an IPv4-only remote host will no longer be accessible. Instead, it will continue to work in the case of 464XLAT.

Similar considerations need to be made regarding the usage of a NAT64 WKP vs. NSP (Section 4.5), as they must match the configuration of DNS64. When using foreign DNS, they may not match. If there is a CLAT and the configured foreign DNS is not a DNS64, the network will keep working only if other means of learning the NAT64 prefix are available.

For broadband networks, as described in Section 4.8, the CEs supporting a CLAT function **SHOULD** support DHCPv6-PD [RFC8415] or alternative means for configuring a shorter prefix. The CE **SHOULD** internally reserve one /64 for the stateless NAT46 translation. The operator must

ensure that the customers are allocated prefixes shorter than /64 in order to support this optimization. One way or another, this is not impacting the performance of the operator network.

Operators may follow "Deployment Considerations" (Section 7 of [RFC6877]) for suggestions on how to take advantage of traffic-engineering requirements.

For cellular networks, the considerations regarding DNSSEC may appear to be out of scope because UEs' Operating Systems commonly don't support DNSSEC. However, applications running on them may, or it may be an Operating System "built-in" support in the future. Moreover, if those devices offer tethering, other client devices behind the UE may be doing the validation; hence, proper DNSSEC support by the operator network is relevant.

Furthermore, cellular networks supporting 464XLAT [RFC6877] and "Discovery of the IPv6 Prefix Used for IPv6 Address Synthesis" [RFC7050] allow a progressive IPv6 deployment, with a single Access Point Name (APN) supporting all types of PDP context (IPv4, IPv6, and IPv4v6). This approach allows the network to automatically serve every possible combination of UEs.

If the operator chooses to provide validation for the DNS64 prefix discovery, it must follow the advice from "Validation of Discovered Pref64::/n" (see Section 3.1 of [RFC7050]).

One last consideration is that many networks may have a mix of different complex scenarios at the same time; for example, customers that require 464XLAT and those that don't, customers that require DNS64 and those that don't, etc. In general, the different issues and the approaches described in this document can be implemented at the same time for different customers or parts of the network. That mix of approaches doesn't present any problem or incompatibility; they work well together as a matter of appropriate and differentiated provisioning. In fact, the NAT64/464XLAT approach facilitates an operator offering both cellular and broadband services to have a single IPv4aaS for both networks while differentiating the deployment key decisions to optimize each case. It's even possible to use hybrid CEs that have a main broadband access link and a backup via the cellular network.

In an ideal world, we could safely use DNS64 if the approach proposed in [DNS-DNSSEC] were followed, avoiding the cases where DNSSEC may be broken. However, this will not solve the issues related to DNS privacy and split DNS.

The only 100% safe solution that also resolves all the issues is, in addition to having a CLAT function, not using a DNS64 but instead making sure that the hosts have a built-in address synthesis feature. Operators could manage to provide CEs with the CLAT function; however, the built-in address synthesis feature is out of their control. If the synthesis is provided by either the Operating System (via its DNS resolver API) or the application (via its own DNS resolver) in such way that the prefix used for the NAT64 function is reachable for the host, the problem goes away.

Whenever feasible, using EAM [RFC7757] as indicated in Section 4.9 provides a very relevant optimization, avoiding double translations.

Applications that require incoming connections typically provide a means for that already. However, PCP and EAM, as indicated in Section 4.10, are valid alternatives, even for creating explicit mappings for customers that require them.

6. Deployment of 464XLAT/NAT64 in Enterprise Networks

The recommendations in this document can also be used in enterprise networks, campuses, and other similar scenarios (including managed end-user networks).

This includes scenarios where the NAT64 function (and DNS64 function, if available) are under the control of that network (or can be configured manually according to that network's specific requirements), and there is a need to provide IPv6-only access to any part of that network, or it is IPv6 only connected to third-party networks.

An example is the IETF meeting network itself, where both NAT64 and DNS64 functions are provided, presenting in this case the same issues as per Section 3.1.1. If there is a CLAT function in the IETF network, then there is no need to use DNS64, and it falls under the considerations of Section 3.1.3. Both scenarios have been tested and verified already in the IETF network.

The following figures represent a few of the possible scenarios.

Figure 13 provides an example of an IPv6-only enterprise network connected with a dual stack to the Internet using local NAT64 and DNS64 functions.

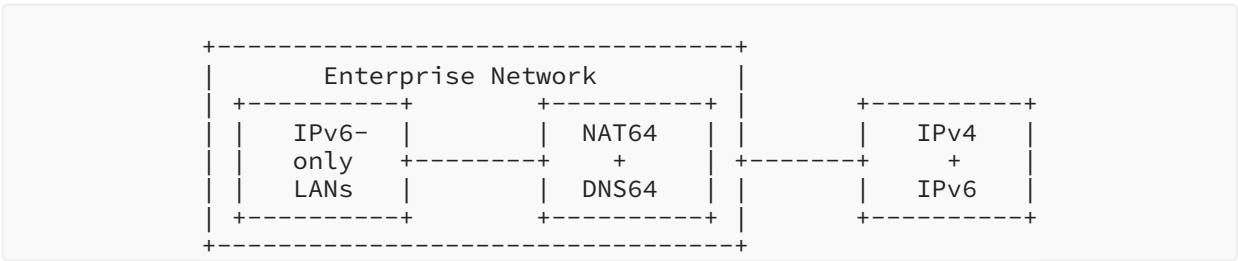

Figure 13: IPv6-Only Enterprise with NAT64 and DNS64

Figure 14 provides an example of a DS enterprise network connected with DS to the Internet using a CLAT function, without a DNS64 function.

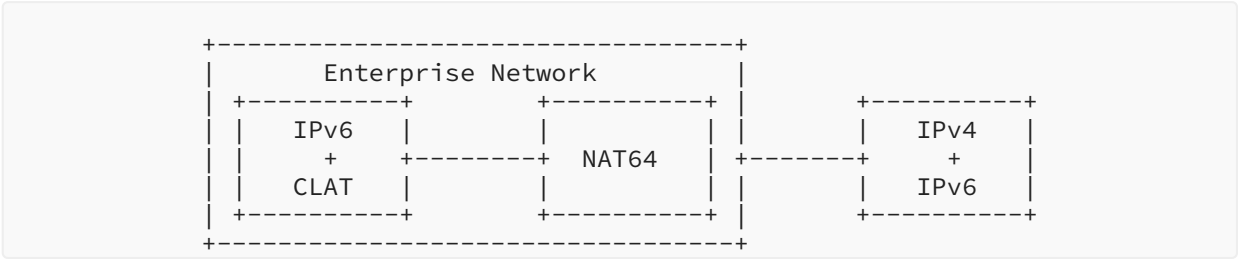

Figure 14: DS Enterprise with CLAT, DS Internet, without DNS64

Finally, Figure 15 provides an example of an IPv6-only provider with a NAT64 function, and a DS enterprise network by means of their own CLAT function, without a DNS64 function.

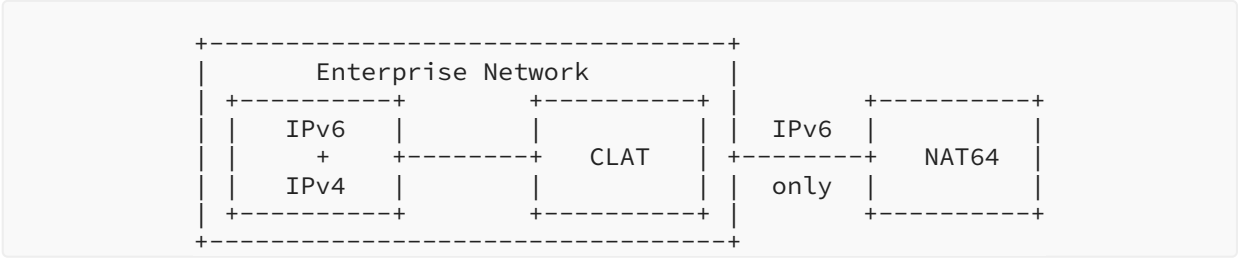

Figure 15: DS Enterprise with CLAT and IPv6-Only Access, without DNS64

7. Security Considerations

This document does not have new specific security considerations beyond those already reported by each of the documents cited. For example, DNS64 [RFC6147] already describes the DNSSEC issues.

As already described in Section 4.4, note that there may be undesirable interactions, especially if using VPNs or DNS privacy, which may impact the correct performance of DNS64/NAT64.

Note that the use of a DNS64 function has privacy considerations that are equivalent to regular DNS, and they are located in either the service provider or an external service provider.

8. IANA Considerations

This document has no IANA actions.

9. References

9.1. Normative References

[RFC1918] Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot, G. J., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, DOI 10.17487/RFC1918, February 1996, <https://www.rfc-editor.org/info/rfc1918>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC5389] Rosenberg, J., Mahy, R., Matthews, P., and D. Wing, "Session Traversal Utilities for NAT (STUN)", RFC 5389, DOI 10.17487/RFC5389, October 2008, <https://www.rfc-editor.org/info/rfc5389>.

[RFC5625] Bellis, R., "DNS Proxy Implementation Guidelines", BCP 152, RFC 5625, DOI 10.17487/RFC5625, August 2009, <https://www.rfc-editor.org/info/rfc5625>.

[RFC5766] Mahy, R., Matthews, P., and J. Rosenberg, "Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN)", RFC 5766, DOI 10.17487/RFC5766, April 2010, <https://www.rfc-editor.org/info/rfc5766>.

[RFC6052] Bao, C., Huitema, C., Bagnulo, M., Boucadair, M., and X. Li, "IPv6 Addressing of IPv4/IPv6 Translators", RFC 6052, DOI 10.17487/RFC6052, October 2010, <https://www.rfc-editor.org/info/rfc6052>.

[RFC6144] Baker, F., Li, X., Bao, C., and K. Yin, "Framework for IPv4/IPv6 Translation", RFC 6144, DOI 10.17487/RFC6144, April 2011, <https://www.rfc-editor.org/info/rfc6144>.

[RFC6146] Bagnulo, M., Matthews, P., and I. van Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers", RFC 6146, DOI 10.17487/RFC6146, April 2011, <https://www.rfc-editor.org/info/rfc6146>.

[RFC6147] Bagnulo, M., Sullivan, A., Matthews, P., and I. van Beijnum, "DNS64: DNS Extensions for Network Address Translation from IPv6 Clients to IPv4 Servers", RFC 6147, DOI 10.17487/RFC6147, April 2011, <https://www.rfc-editor.org/info/rfc6147>.

[RFC6535] Huang, B., Deng, H., and T. Savolainen, "Dual-Stack Hosts Using "Bump-in-the-Host" (BIH)", RFC 6535, DOI 10.17487/RFC6535, February 2012, <https://www.rfc-editor.org/info/rfc6535>.

[RFC6877] Mawatari, M., Kawashima, M., and C. Byrne, "464XLAT: Combination of Stateful and Stateless Translation", RFC 6877, DOI 10.17487/RFC6877, April 2013, <https://www.rfc-editor.org/info/rfc6877>.

[RFC6887] Wing, D., Ed., Cheshire, S., Boucadair, M., Penno, R., and P. Selkirk, "Port Control Protocol (PCP)", RFC 6887, DOI 10.17487/RFC6887, April 2013, <https://www.rfc-editor.org/info/rfc6887>.

[RFC7050] Savolainen, T., Korhonen, J., and D. Wing, "Discovery of the IPv6 Prefix Used for IPv6 Address Synthesis", RFC 7050, DOI 10.17487/RFC7050, November 2013, <https://www.rfc-editor.org/info/rfc7050>.

[RFC7225] Boucadair, M., "Discovering NAT64 IPv6 Prefixes Using the Port Control Protocol (PCP)", RFC 7225, DOI 10.17487/RFC7225, May 2014, <https://www.rfc-editor.org/info/rfc7225>.

[RFC7757] Anderson, T. and A. Leiva Popper, "Explicit Address Mappings for Stateless IP/ICMP Translation", RFC 7757, DOI 10.17487/RFC7757, February 2016, <https://www.rfc-editor.org/info/rfc7757>.

[RFC7915] Bao, C., Li, X., Baker, F., Anderson, T., and F. Gont, "IP/ICMP Translation Algorithm", RFC 7915, DOI 10.17487/RFC7915, June 2016, <https://www.rfc-editor.org/info/rfc7915>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8273] Brzozowski, J. and G. Van de Velde, "Unique IPv6 Prefix per Host", RFC 8273, DOI 10.17487/RFC8273, December 2017, <https://www.rfc-editor.org/info/rfc8273>.

[RFC8305] Schinazi, D. and T. Pauly, "Happy Eyeballs Version 2: Better Connectivity Using Concurrency", RFC 8305, DOI 10.17487/RFC8305, December 2017, <https://www.rfc-editor.org/info/rfc8305>.

[RFC8375] Pfister, P. and T. Lemon, "Special-Use Domain 'home.arpa.'", RFC 8375, DOI 10.17487/RFC8375, May 2018, <https://www.rfc-editor.org/info/rfc8375>.

[RFC8415] Mrugalski, T., Siodelski, M., Volz, B., Yourtchenko, A., Richardson, M., Jiang, S., Lemon, T., and T. Winters, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 8415, DOI 10.17487/RFC8415, November 2018, <https://www.rfc-editor.org/info/rfc8415>.

[RFC8445] Keranen, A., Holmberg, C., and J. Rosenberg, "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal", RFC 8445, DOI 10.17487/RFC8445, July 2018, <https://www.rfc-editor.org/info/rfc8445>.

[RFC8484] Hoffman, P. and P. McManus, "DNS Queries over HTTPS (DoH)", RFC 8484, DOI 10.17487/RFC8484, October 2018, <https://www.rfc-editor.org/info/rfc8484>.

9.2. Informative References

[About-DNS64] Linkova, J., "Let's talk about IPv6 DNS64 & DNSSEC", June 2016, <https://blog.apnic.net/2016/06/09/lets-talk-ipv6-dns64-dnssec/>.

[ARCEP] ARCEP, "Service client des operateurs : les mesures de qualite de service", April 2018, <https://www.arcep.fr/cartes-et-donnees/nos-publications-chiffrees/service-client-des-operateurs-mesures-de-la-qualite-de-service/service-client-des-operateurs-les-mesures-de-qualite-de-service.html>.

[DHCPv6-OPTIONS] Li, L., Cui, Y., Liu, C., Wu, J., Baker, F., and J. Palet, "DHCPv6 Options for Discovery NAT64 Prefixes", Work in Progress, Internet-Draft, draft-li-intarea-nat64-prefix-dhcp-option-02, 20 April 2019, <https://tools.ietf.org/html/draft-li-intarea-nat64-prefix-dhcp-option-02>.

[DNS-DNSSEC] Byrne, C. and J. Palet, "IPv6-Ready DNS/DNSSSEC Infrastructure", Work in Progress, Internet-Draft, draft-bp-v6ops-ipv6-ready-dns-dnssec-00, 10 October 2018, <https://tools.ietf.org/html/draft-bp-v6ops-ipv6-ready-dns-dnssec-00>.

[DNS-RPZ] Vixie, P. and V. Schryver, "DNS Response Policy Zones (RPZ)", Work in Progress, Internet-Draft, draft-vixie-dnsop-dns-rpz-00, 23 June 2018, <https://tools.ietf.org/html/draft-vixie-dnsop-dns-rpz-00>.

[DNS64-Benchm] Lencse, G. and Y. Kadobayashi, "Benchmarking DNS64 Implementations: Theory and Practice", pp. 61-74, no. 1, vol. 127, Computer Communications, DOI 10.1016/j.comcom.2018.05.005, September 2018, <https://www.sciencedirect.com/science/article/pii/S0140366418302184?via%3Dihub>.

[DNS64-BM-Meth] Lencse, G., Georgescu, M., and Y. Kadobayashi, "Benchmarking Methodology for DNS64 Servers", pp. 162-175, no. 1, vol. 109, Computer Communications, DOI 10.1016/j.comcom.2017.06.004, September 2017, <https://www.sciencedirect.com/science/article/pii/S0140366416305904?via%3Dihub>.

[FCC] FCC, "Measuring Broadband America Mobile 2013-2018 Coarsened Data", December 2018, <https://www.fcc.gov/reports-research/reports/measuring-broadband-america/measuring-broadband-america-mobile-2013-2018>.

[IPV4ONLY-ARPA] Cheshire, S. and D. Schinazi, "Special Use Domain Name 'ipv4only.arpa'", Work in Progress, Internet-Draft, draft-cheshire-sudn-ipv4only-dot-arpa-14, 3 November 2018, <https://tools.ietf.org/html/draft-cheshire-sudn-ipv4only-dot-arpa-14>.

[IPv6-TRANSITION] Lencse, G., Palet, J., Howard, L., Patterson, R., and I. Farrer, "Pros and Cons of IPv6 Transition Technologies for IPv4aaS", Work in Progress, Internet-Draft, draft-lmhp-v6ops-transition-comparison-03, 6 July 2019, <https://tools.ietf.org/html/draft-lmhp-v6ops-transition-comparison-03>.

[ISOC] ISOC, "State of IPv6 Deployment 2018", June 2018, <https://www.internetsociety.org/resources/2018/state-of-ipv6-deployment-2018/>.

[OPT-464XLAT] Palet, J. and A. D'Egidio, "464XLAT Optimization", Work in Progress, Internet-Draft, draft-palet-v6ops-464xlat-opt-cdn-caches-03, 8 July 2019, <https://tools.ietf.org/html/draft-palet-v6ops-464xlat-opt-cdn-caches-03>.

[PREF64] Colitti, L. and J. Linkova, "Discovering PREF64 in Router Advertisements", Work in Progress, Internet-Draft, draft-ietf-6man-ra-pref64-06, 3 October 2019, <https://tools.ietf.org/html/draft-ietf-6man-ra-pref64-06>.

[QUIC-CONNECTIONS] Huitema, C., Shore, M., Mankin, A., Dickinson, S., and J. Iyengar, "Specification of DNS over Dedicated QUIC Connections", Work in Progress, Internet-Draft, draft-huitema-quic-dnsoquic-07, 7 September 2019, <https://tools.ietf.org/html/draft-huitema-quic-dnsoquic-07>.

[RFC6889] Penno, R., Saxena, T., Boucadair, M., and S. Sivakumar, "Analysis of Stateful 64 Translation", RFC 6889, DOI 10.17487/RFC6889, April 2013, <https://www.rfc-editor.org/info/rfc6889>.

[RFC6950] Peterson, J., Kolkman, O., Tschofenig, H., and B. Aboba, "Architectural Considerations on Application Features in the DNS", RFC 6950, DOI 10.17487/RFC6950, October 2013, <https://www.rfc-editor.org/info/rfc6950>.

[RFC7051]

Korhonen, J., Ed. and T. Savolainen, Ed., "Analysis of Solution Proposals for Hosts to Learn NAT64 Prefix", RFC 7051, DOI 10.17487/RFC7051, November 2013, <https://www.rfc-editor.org/info/rfc7051>.

[RFC7269] Chen, G., Cao, Z., Xie, C., and D. Binet, "NAT64 Deployment Options and Experience", RFC 7269, DOI 10.17487/RFC7269, June 2014, <https://www.rfc-editor.org/info/rfc7269>.

[RFC7755] Anderson, T., "SIIT-DC: Stateless IP/ICMP Translation for IPv6 Data Center Environments", RFC 7755, DOI 10.17487/RFC7755, February 2016, <https://www.rfc-editor.org/info/rfc7755>.

[RFC7756] Anderson, T. and S. Steffann, "Stateless IP/ICMP Translation for IPv6 Internet Data Center Environments (SIIT-DC): Dual Translation Mode", RFC 7756, DOI 10.17487/RFC7756, February 2016, <https://www.rfc-editor.org/info/rfc7756>.

[RFC7849] Binet, D., Boucadair, M., Vizdal, A., Chen, G., Heatley, N., Chandler, R., Michaud, D., Lopez, D., and W. Haeffner, "An IPv6 Profile for 3GPP Mobile Devices", RFC 7849, DOI 10.17487/RFC7849, May 2016, <https://www.rfc-editor.org/info/rfc7849>.

[RFC7858] Hu, Z., Zhu, L., Heidemann, J., Mankin, A., Wessels, D., and P. Hoffman, "Specification for DNS over Transport Layer Security (TLS)", RFC 7858, DOI 10.17487/RFC7858, May 2016, <https://www.rfc-editor.org/info/rfc7858>.

[RFC8094] Reddy, T., Wing, D., and P. Patil, "DNS over Datagram Transport Layer Security (DTLS)", RFC 8094, DOI 10.17487/RFC8094, February 2017, <https://www.rfc-editor.org/info/rfc8094>.

[RFC8219] Georgescu, M., Pislaru, L., and G. Lencse, "Benchmarking Methodology for IPv6 Transition Technologies", RFC 8219, DOI 10.17487/RFC8219, August 2017, <https://www.rfc-editor.org/info/rfc8219>.

[RFC8585] Palet Martinez, J., Liu, H. M.-H., and M. Kawashima, "Requirements for IPv6 Customer Edge Routers to Support IPv4-as-a-Service", RFC 8585, DOI 10.17487/RFC8585, May 2019, <https://www.rfc-editor.org/info/rfc8585>.

[RIPE-690] RIPE, "Best Current Operational Practice for Operators: IPv6 prefix assignment for end-users - persistent vs non-persistent, and what size to choose", October 2017, <https://www.ripe.net/publications/docs/ripe-690>.

[Threat-DNS64] Lencse, G. and Y. Kadobayashi, "Methodology for the identification of potential security issues of different IPv6 transition technologies: Threat analysis of DNS64 and stateful NAT64", pp. 397-411, no. 1, vol. 77, Computers & Security, DOI 10.1016/j.cose.2018.04.012, August 2018, <https://www.sciencedirect.com/science/article/pii/S0167404818303663?via%3Dihub>.

Appendix A. Example of Broadband Deployment with 464XLAT

This section summarizes how an operator may deploy an IPv6-only network for residential/SOHO customers, supporting IPv6 inbound connections, and IPv4-as-a-Service (IPv4aaS) by using 464XLAT.

Note that an equivalent setup could also be provided for enterprise customers. If they need to support IPv4 inbound connections, several mechanisms, depending on specific customer needs, allow it; see [RFC7757].

Conceptually, most of the operator network could be IPv6 only (represented in the next figures as "IPv6-only flow"), or even if part of the network is actually dual stack, only IPv6 access is available for some customers (i.e., residential customers). This part of the network connects the IPv6-only subscribers (by means of IPv6-only access links) to the IPv6 upstream providers and to the IPv4-Internet by means of NAT64 (PLAT in the 464XLAT terminology).

The traffic flow from and back to the CE to services available in the IPv6 Internet (or even dual-stack remote services, when IPv6 is being used) is purely native IPv6 traffic, so there are no special considerations about it.

From the DNS perspective, there are remote networks with IPv4 only that will typically have only IPv4 DNS (DNS/IPv4) or will at least be seen as IPv4 DNS from the CE perspective. On the operator side, the DNS, as seen from the CE, is only IPv6 (DNS/IPv6), and it also has a DNS64 function.

On the customer LANs side, there is actually one network, which of course could be split into different segments. The most common setup will be dual-stack segments, using global IPv6 addresses and [RFC1918] for IPv4, in any regular residential / Small Office, Home Office (SOHO) IPv4 network. In the figure below, it is represented as tree segments to show that the three possible setups are valid (IPv6 only, IPv4 only, and dual stack).

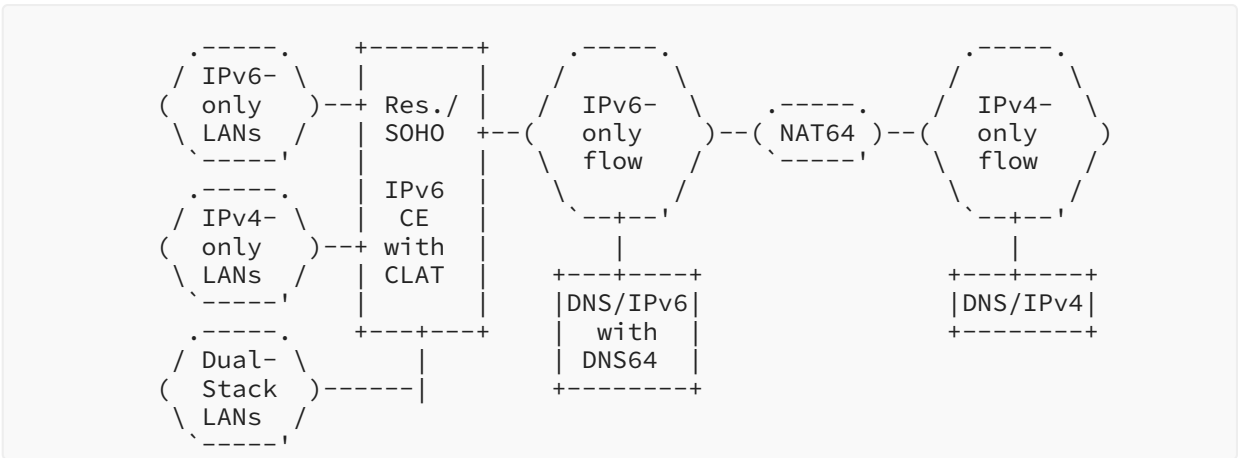

Figure 16: CE Setup with Built-In CLAT, with DNS64

In addition to the regular CE setup, which typically will be access-technology dependent, the steps for the CLAT function configuration can be summarized as follows:

1. Discovery of the PLAT (NAT64) prefix: It may be done using [RFC7050], [RFC7225] in those networks where PCP is supported, or other alternatives that may be available in the future, such as Router Advertising [PREF64] or DHCPv6 options [DHCPv6-OPTIONS].
2. If the CLAT function allows stateless NAT46 translation, a /64 from the pool typically provided to the CE by means of DHCPv6-PD [RFC8415] needs to be set aside for that translation. Otherwise, the CLAT is forced to perform an intermediate stateful NAT44 before the stateless NAT46, as described in Section 4.8.

A more detailed configuration approach is described in [RFC8585].

The operator network needs to ensure that the correct responses are provided for the discovery of the PLAT prefix. It is highly recommended that [RIPE-690] be followed in order to ensure that multiple /64s are available, including the one needed for the NAT46 stateless translation.

The operator needs to understand other issues, as described throughout this document, in order to make relevant decisions. For example, if several NAT64 functions are needed in the context of scalability / high availability, an NSP should be considered (see Section 4.5).

More complex scenarios are possible, for example, if a network offers multiple NAT64 prefixes, destination-based NAT64 prefixes, etc.

If the operator decides not to provide a DNS64 function, then this setup will be the same as the following figure. This will also be the setup that will be seen from the perspective of the CE, if a foreign DNS is used and consequently is not the operator-provided DNS64 function.

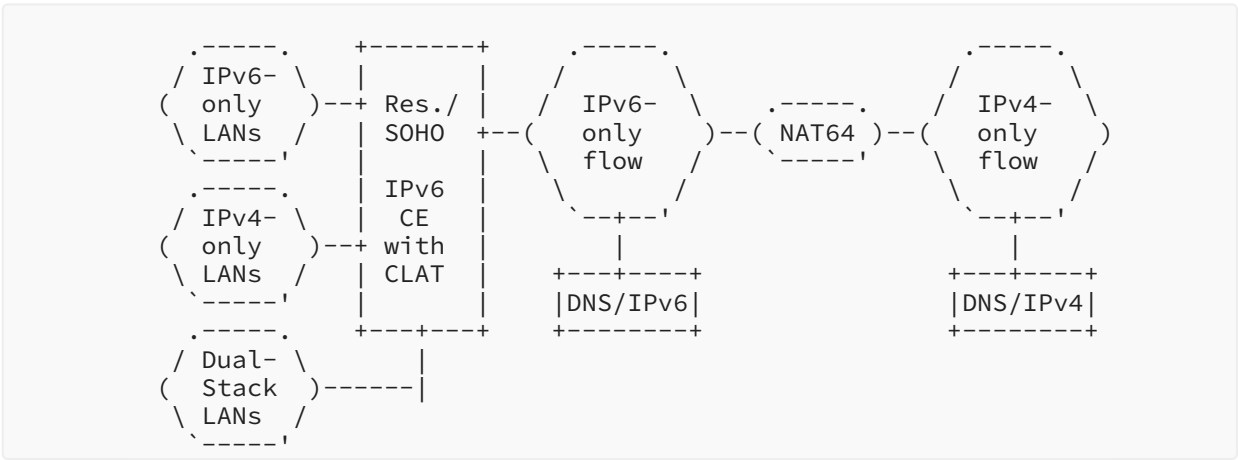

Figure 17: CE Setup with Built-In CLAT, without DNS64

In this case, the discovery of the PLAT prefix needs to be arranged as indicated in Section 4.1.1.

In addition, if the CE doesn't have a built-in CLAT function, the customer can choose to set up the IPv6 operator-managed CE in bridge mode (and optionally use an external router). Or, for example, if there is an access technology that requires some kind of media converter (Optical Network Termination (ONT) for fiber to the home (FTTH), Cable Modem for Data-Over-Cable Service Interface Specification (DOCSIS), etc.), the complete setup will look like Figure 18. Obviously, there will be some intermediate configuration steps for the bridge, depending on the specific access technology/protocols, which should not modify the steps already described in the previous cases for the CLAT function configuration.

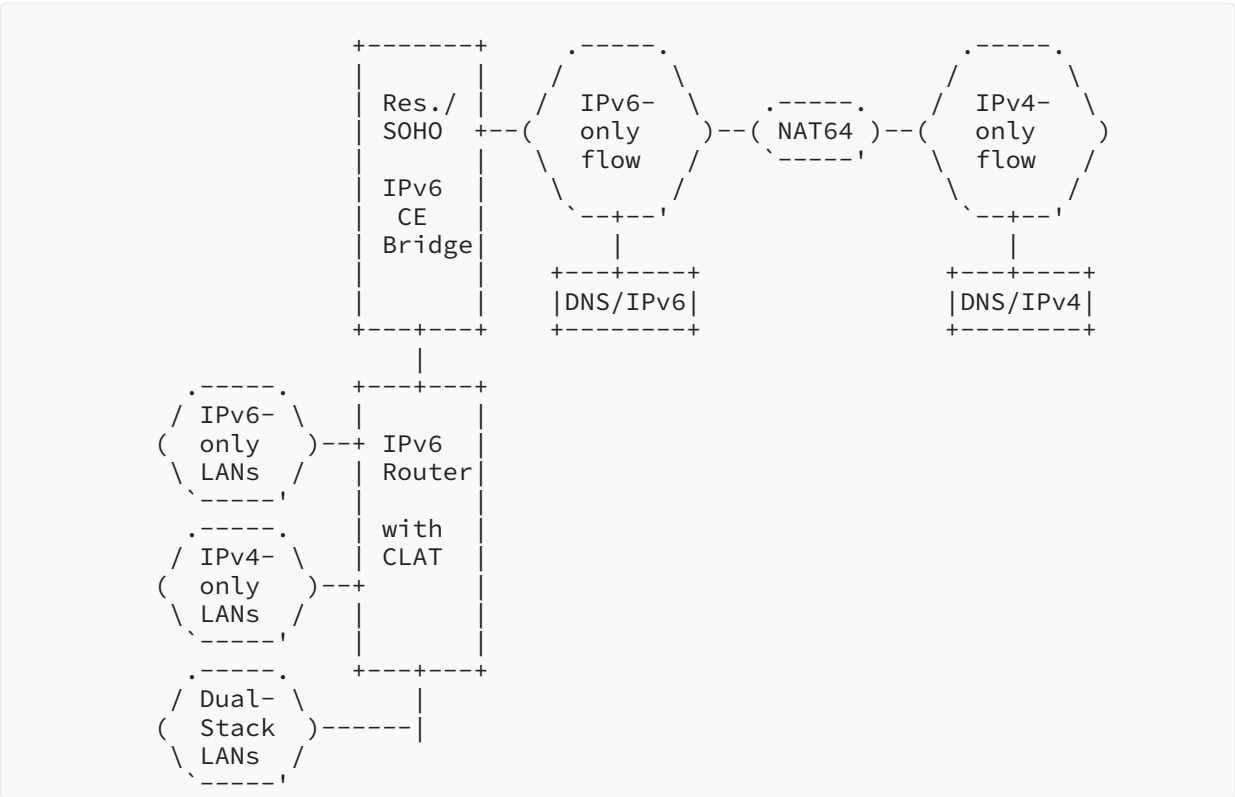

Figure 18: CE Setup with Bridged CLAT, without DNS64

Several routers (i.e., the operator-provided CE and the downstream user-provided router) that enable simultaneous routing and/or CLAT should be avoided to ensure that multiple NAT44 and NAT46 levels are not used and that the operation of multiple IPv6 subnets is correct. In those cases, the use of the Home Networking Control Protocol (HNCP) [RFC8375] is suggested.

Note that the procedure described here for the CE setup can be simplified if the CE follows [RFC8585].

Appendix B. CLAT Implementation

In addition to the regular set of features for a CE, a CLAT CE implementation requires support for:

- [RFC7915] for the NAT46 function.
- [RFC7050] for the PLAT prefix discovery.
- [RFC7225] for the PLAT prefix discovery if PCP is supported.
- [PREF64] for the PLAT prefix discovery by means of Router Advertising.
- [DHCPv6-OPTIONS] for the PLAT prefix discovery by means of DHCP.
- If stateless NAT46 is supported, a mechanism to ensure that multiple /64 are available, such as DHCPv6-PD [RFC8415], must be used.

There are several Open Source implementations of CLAT, such as:

- Android: https://github.com/ddrown/android_external_android-clat
- Jool: https://www.jool.mx
- Linux: https://github.com/toreanderson/clatd
- OpenWRT: https://git.openwrt.org/?p=openwrt%2Fopenwrt.git&a=search&h=refs%2Ftags%2Fv19.07.0-rc1&st=commit&s=464xlat
- VPP: https://git.fd.io/vpp/tree/src/plugins/nat

Appendix C. Benchmarking

A benchmarking methodology for IPv6 transition technologies has been defined in [RFC8219]. NAT64 and 464XLAT are addressed among the single- and double-translation technologies, respectively. DNS64 is addressed in Section 9, and the methodology is elaborated in [DNS64-BM-Meth] of that document.

Several documents provide references to benchmarking results, for example, for DNS64 [DNS64-Benchm].

Acknowledgements

The author would like to acknowledge the inputs of Gabor Lencse, Andrew Sullivan, Lee Howard, Barbara Stark, Fred Baker, Mohamed Boucadair, Alejandro D'Egidio, Dan Wing, Mikael Abrahamsson, and Eric Vyncke.

Conversations with Marcelo Bagnulo, one of the coauthors of NAT64 and DNS64, and email correspondence via the IETF mailing lists with Mark Andrews have been very useful for this work.

Work on this document was inspired by Christian Huitema, who suggested that DNS64 should never be used when deploying CLAT in the IETF network.

Author's Address

Jordi Palet Martinez
The IPv6 Company
Molino de la Navata, 75
28420 La Navata - Galapagar Madrid
Spain
Email: jordi.palet@theipv6company.com
URI: http://www.theipv6company.com/